

**We claim:**

1. A method for analyzing reuse patterns of accesses of data by a program running on a computing device, the computing device having a memory in which the data are stored and from which the data are accessed, the method comprising:

- 5           (a) running the program on the computing device;
- (b) monitoring the accesses of the data by the program during step (a); and
- (c) determining a reuse distance for each datum from among the data accessed by the program during step (a), the reuse distance being a number of distinct data which are accessed between two accesses of the datum.

10           2. The method of claim 1, wherein step (c) comprises:

             determining a last access time of each of the data;

             organizing a search tree from the last accesses, wherein the search tree comprises a node for each of the data, the node comprising the last access time and a weight of a sub-tree of the node; and

15           compressing the search tree in accordance with a bounded relative error.

             3. The method of claim 2, wherein the search tree is compressed by (i) determining a capacity of each node in accordance with the reuse distance and the bounded relative error and (ii) merging adjacent ones of the nodes in accordance with the capacities of the nodes.

             4. The method of claim 1, wherein step (c) comprises:

20           determining a last access time of each of the data;

             maintaining a trace storing the last access times of the last  $C$  accesses of the data, where  $C$  is a cut-off distance; and

maintaining a search tree storing access times other than the last  $C$  accesses, each node in the search tree having a capacity  $B$ , where  $B$  is a bounded absolute error.

5. The method of claim 1, further comprising:

(d) determining a reuse pattern from the reuse distances determined in step (c).

6. The method of claim 5, wherein step (d) comprises forming a reuse distance histogram of the reuse distances by absolute ranges of the reuse distances.

7. The method of claim 6, wherein step (d) further comprises forming a reference histogram of the reuse distances by percentile ranges of the reuse distances.

8. The method of claim 7, wherein the reference histogram is formed for a plurality of training inputs.

9. The method of claim 8, wherein step (d) further comprises using the reference histograms for the plurality of training inputs to map data size to the reuse distance.

10. The method of claim 9, wherein the data size is mapped to the reuse distance through linear fitting.

11. The method of claim 6, further comprising:

(e) from the reuse distance histogram, forming an affinity group of at least two data which are always accessed within a distance  $k$  of one another, wherein  $k$  is a predetermined quantity.

12. The method of claim 11, wherein step (e) comprises selecting the data in the affinity group such that the data in the affinity group have average reuse distances which fulfill a necessary condition with respect to  $k$ .

13. The method of claim 15, wherein the necessary condition is that the average reuse distances differ by no more than  $k$ .

14. The method of claim 12, wherein:

the reuse distance histogram comprises  $B$  bins; and

the necessary condition is that differences between the average reuse distances, summed over all of the bins, do not exceed  $kB$ .

5        15. The method of claim 14, wherein step (e) comprises:

(i) initially treating each of the data as an affinity group;

(ii) traversing all of the affinity groups and merging any two affinity groups for which the necessary condition is met; and

(iii) performing step (e)(ii) until no more of the affinity groups can be merged.

10       16. The method of claim 11, wherein step (e) is performed a plurality of times for different values of  $k$ .

17. The method of claim 1, further comprising:

comparing reuse signatures of the data to determine whether two or more of the data have reuse signatures which differ by less than a predetermined percentage; and

15       for any two or more of the data whose reuse signatures differ by less than said predetermined percentage, identifying a reference affinity among said two or more data.

18. A computing device capable of analyzing reuse patterns of accesses of data by a program running on a computing device, the computing device comprising:

a memory in which the data are stored and from which the data are accessed; and

20       a processor, in communication with the memory, for:

(a) running the program on the computing device;

(b) monitoring the accesses of the data by the program during step (a); and

(c) determining a reuse distance for each datum from among the data accessed by the program during step (a), the reuse distance being a number of distinct data which are accessed between two accesses of the datum.

19. The computing device of claim 18, wherein the processor performs step (c) by:

5 determining a last access time of each of the data;

organizing a search tree from the last accesses, wherein the search tree comprises a node for each of the data, the node comprising the last access time and a weight of a sub-tree of the node; and

compressing the search tree in accordance with a bounded relative error.

10 20. The computing device of claim 19, wherein the search tree is compressed by (i) determining a capacity of each node in accordance with the reuse distance and the bounded relative error and (ii) merging adjacent ones of the nodes in accordance with the capacities of the nodes.

21. The computing device of claim 18, wherein the processor performs step (c) by:

15 maintaining a trace storing the last access times of the last  $C$  accesses of the data, where  $C$  is a cut-off distance; and

maintaining a search tree storing access times other than the last  $C$  accesses, each node in the search tree having a capacity  $B$ , where  $B$  is a bounded absolute error.

22. The computing device of claim 18, wherein the processor further performs:

20 (d) determining a reuse pattern from the reuse distances determined in step (c).

23. The computing device of claim 22, wherein the processor performs step (d) by forming a reuse distance histogram of the reuse distances by absolute ranges of the reuse distances.

24. The computing device of claim 23, wherein the processor further performs step (d) by forming a reference histogram of the reuse distances by percentile ranges of the reuse distances.

25. The computing device of claim 24, wherein the reference histogram is formed for a plurality of training inputs.

26. The computing device of claim 25, wherein the processor performs step (d) further by using the reference histograms for the plurality of training inputs to map data size to the reuse distance.

27. The computing device of claim 26, wherein the data size is mapped to the reuse distance through linear fitting.

28. The computing device of claim 23, wherein the processor further performs:

(e) from the reuse distance histogram, forming an affinity group of at least two data which are always accessed within a distance  $k$  of one another, wherein  $k$  is a predetermined quantity.

29. The computing device of claim 28, wherein the processor performs step (e) by selecting the data in the affinity group such that the data in the affinity group have average reuse distances which fulfill a necessary condition with respect to  $k$ .

30. The computing device of claim 29, wherein the necessary condition is that the average reuse distances differ by no more than  $k$ .

31. The computing device of claim 29, wherein:

the reuse distance histogram comprises  $B$  bins; and  
the necessary condition is that differences between the average reuse distances, summed over all of the bins, do not exceed  $kB$ .

32. The computing device of claim 31, wherein the processor performs step (e) by:

(i) initially treating each of the data as an affinity group;

(ii) traversing all of the affinity groups and merging any two affinity groups for which the necessary condition is met; and

(iii) performing step (e)(ii) until no more of the affinity groups can be merged.

5           33. The computing device of claim 32, wherein step (e) is performed a plurality of times for different values of  $k$ .

34. The computing device of claim 18, wherein the processing device further performs:

comparing reuse signatures of the data to determine whether two or more of the data have reuse signatures which differ by less than a predetermined percentage; and

10           for any two or more of the data whose reuse signatures differ by less than said predetermined percentage, identifying a reference affinity among said two or more data.

35. A method for analyzing affinities among a plurality of events, the method comprising:

(a) monitoring occurrences of the events;

15           (b) determining a reoccurrence distance for each event, the reoccurrence distance being a number of distinct ones of the plurality of events which occur between two occurrences of said each event; and

(c) determining, from the reoccurrence distance determined in step (b), an affinity among at least two of the events, the affinity being a tendency of said at least two of the events to occur  
20 together.

36. The method of claim 35, wherein step (c) comprises determining the affinity such that said events always occur within a distance  $k$  of each other, wherein  $k$  is a predetermined quantity

and the distance is a number of distinct events occurring between occurrences of said at least two of the events.

37. The method of claim 35, wherein step (c) comprises comparing reoccurrence signatures of the events to determine whether said two or more of the events have reoccurrence  
5 signatures which differ by less than a predetermined percentage.